Hamiltonian Cycle

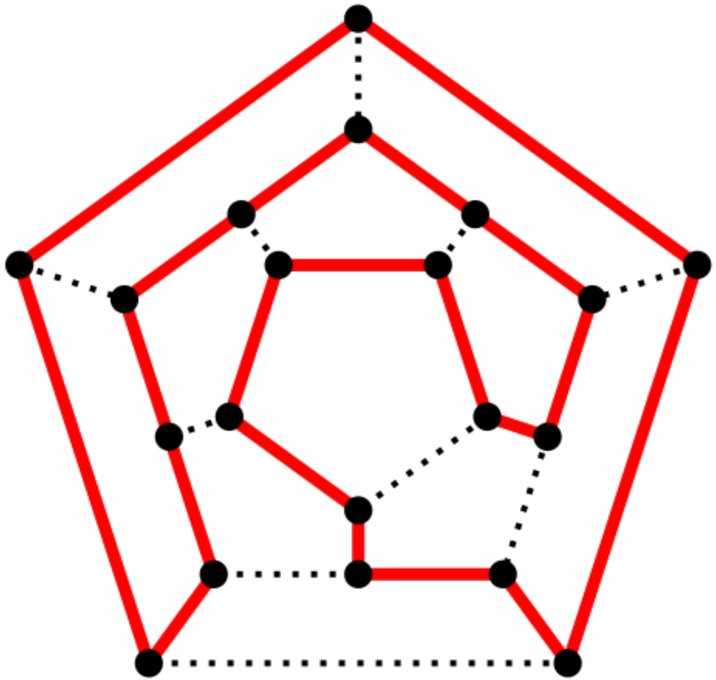
Zero Knowledge Proof

Hamiltonian cycle

Hamiltonian cycle

- A path that visits each vertex exactly once, and ends at the same point it started

Example



Hamiltonian cycle

- A path that visits each vertex exactly once, and ends at the same point it started
- William Rowan Hamilton (1805-1865)

Hamiltonian cycle

- A path that visits each vertex exactly once, and ends at the same point it started
- William Rowan Hamilton (1805-1865)

Hamiltonian cycle

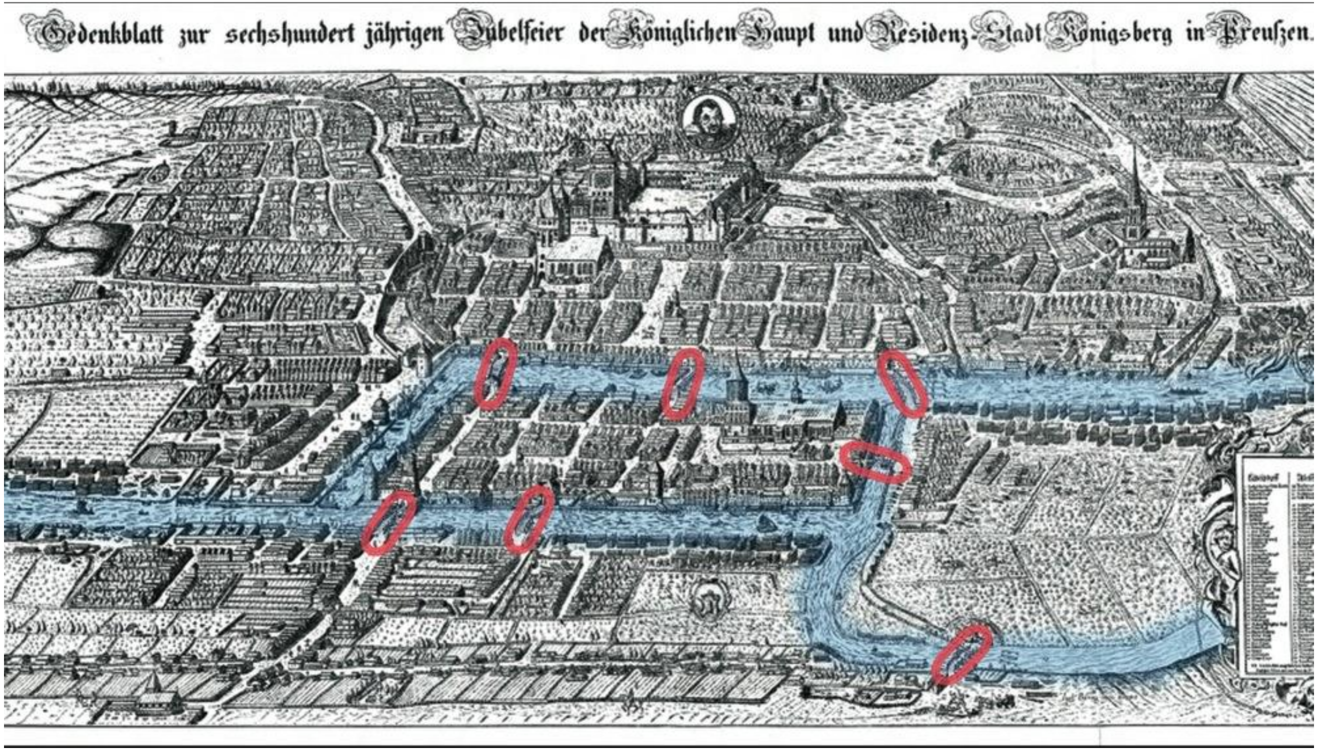
- A path that visits each vertex exactly once, and ends at the same point it started
- William Rowan Hamilton (1805-1865)

Eulerian path/cycle

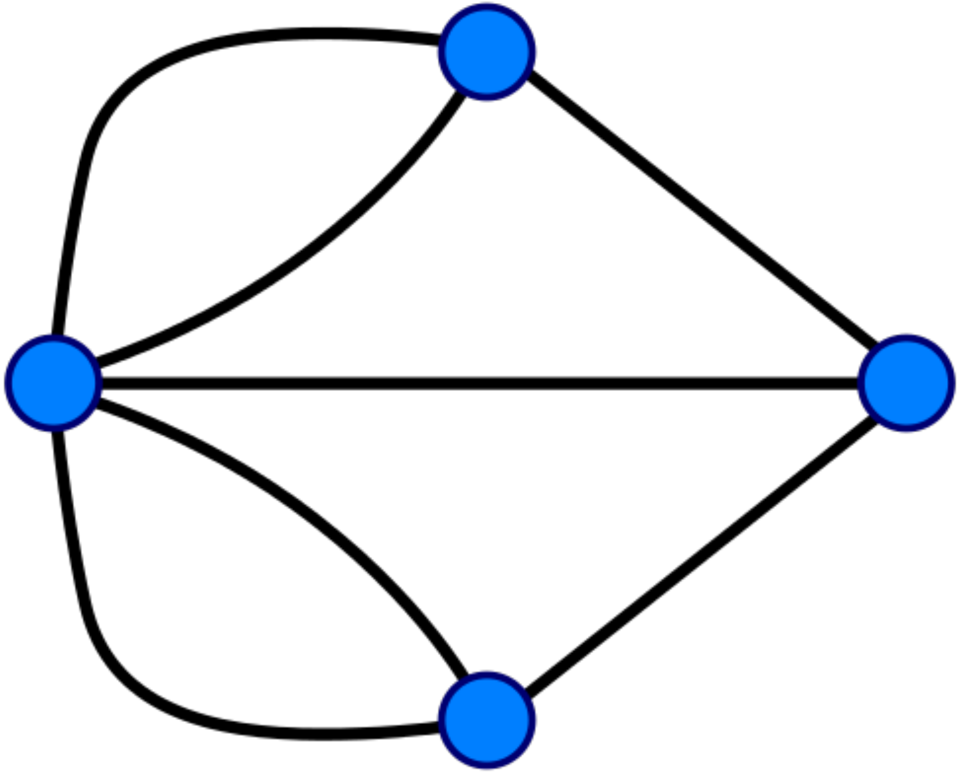
Eulerian path/cycle

- Seven Bridges of Köningsberg

Seven Bridges



Seven Bridges



Eulerian path/cycle

- Seven Bridges of Köningsberg
- Eulerian path: visits each edge exactly once

Eulerian path/cycle

- Seven Bridges of Köningsberg
- Eulerian path: visits each edge exactly once
- Eulerian cycle: starts and ends at the same point

Eulerian path/cycle

- Seven Bridges of Köningsberg
- Eulerian path: visits each edge exactly once
- Eulerian cycle: starts and ends at the same point
- Graph has Eulerian circuit iff (1) connected and (2) all vertices have even degree.

Complexity

Complexity

- Euler vs. Hamilton

Complexity

- Euler vs. Hamilton
- Edges vs. Vertices

Complexity

- Euler vs. Hamilton
- Edges vs. Vertices
- P vs. NP

Complexity

- Euler vs. Hamilton
- Edges vs. Vertices
- P vs. NP
- No necessary and sufficient conditions for a Hamiltonian cycle

Complexity

- Euler vs. Hamilton
- Edges vs. Vertices
- P vs. NP
- No necessary and sufficient conditions for a Hamiltonian cycle
- No good algorithm for finding one (there are known algorithms with running time $O(n^2 2^n)$ and $O(1.657^n)$), so exponential

Zero Knowledge (1)

Zero Knowledge (1)

- The problem: Suppose that P knows a Hamiltonian Cycle for a graph G. How can she prove this to V in zero-knowledge?

Zero Knowledge (1)

- The problem: Suppose that P knows a Hamiltonian Cycle for a graph G. How can she prove this to V in zero-knowledge?
- Difference cycle and Hamiltonian cycle

Zero Knowledge (1)

- The problem: Suppose that P knows a Hamiltonian Cycle for a graph G. How can she prove this to V in zero-knowledge?
- Difference cycle and Hamiltonian cycle
- Permuting: Create a graph F that is isomorphic to G

Zero Knowledge (2)

- Step 1: P randomly creates F isomorphic to G

Zero Knowledge (2)

- Step 1: P randomly creates F isomorphic to G
- Step 2: P commits to F (how?)

Zero Knowledge (2)

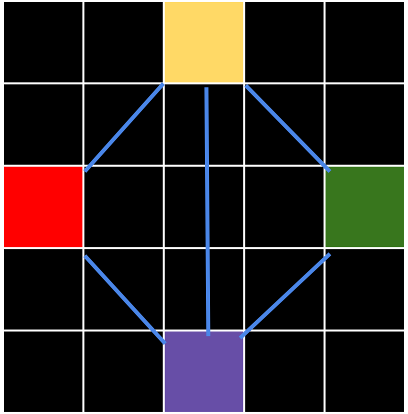
- Step 1: P randomly creates F isomorphic to G
- Step 2: P commits to F (how?)
- Step 3: V chooses between revealing (1) the isomorphism or (2) the Hamiltonian cycle

Zero Knowledge (2)

- Step 1: P randomly creates F isomorphic to G
- Step 2: P commits to F (how?)
- Step 3: V chooses between revealing (1) the isomorphism or (2) the Hamiltonian cycle
- Step 4: P reveals (1) F completely plus the isomorphism or (2) the Hamiltonian cycle

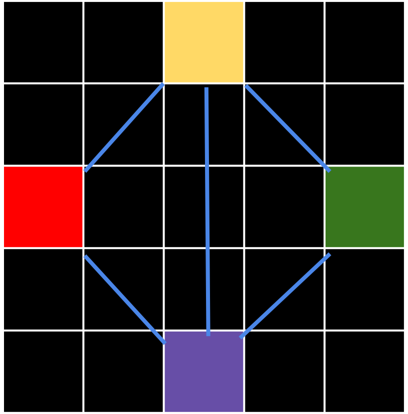
Commitment and example

Commitment and example

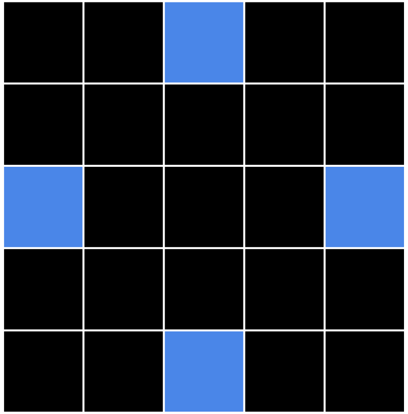


G

Commitment and example

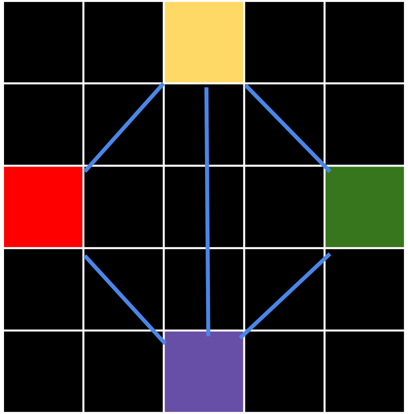


G

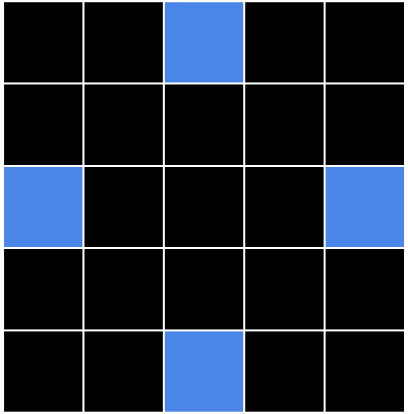


F

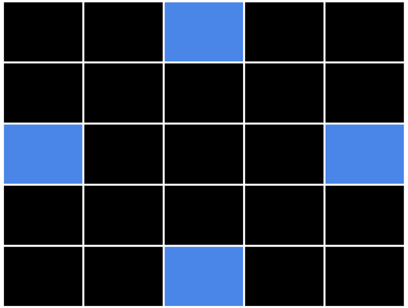
Commitment and example

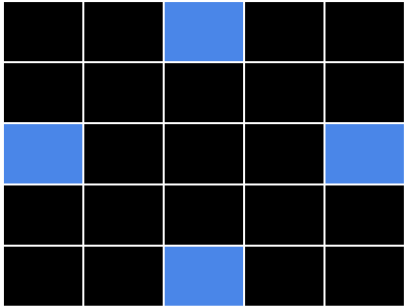


G

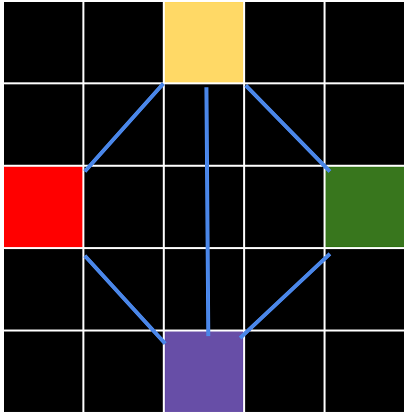


F

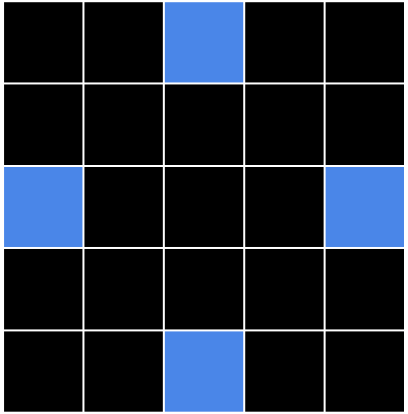




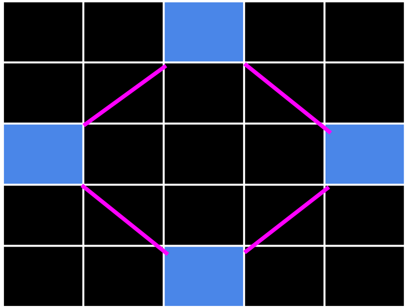
Commitment and example



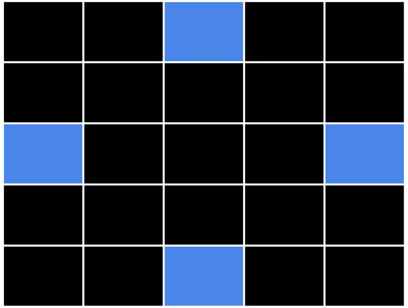
G



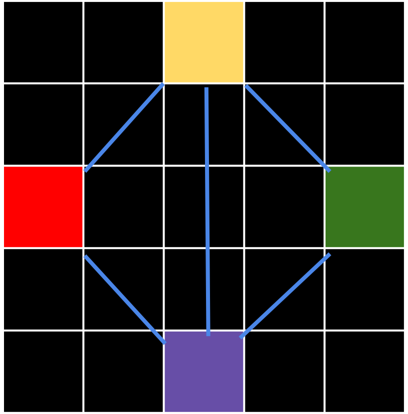
F



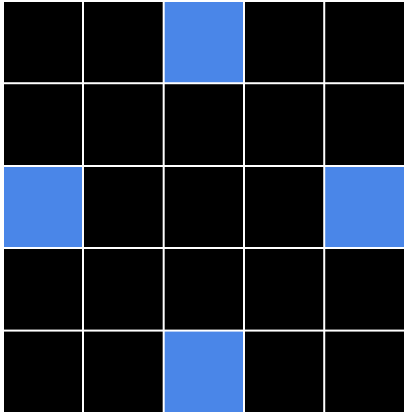
Cycle



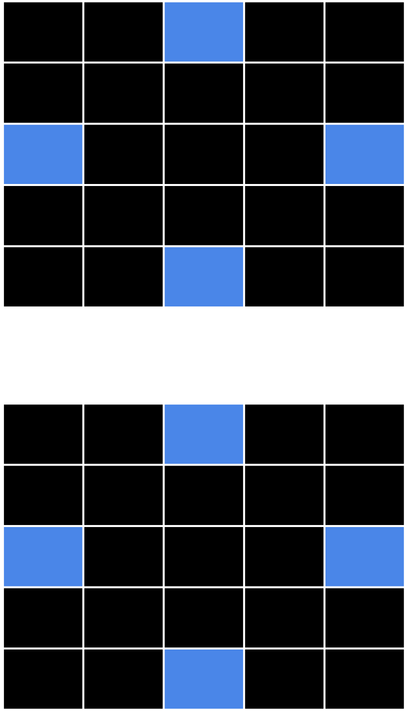
Commitment and example



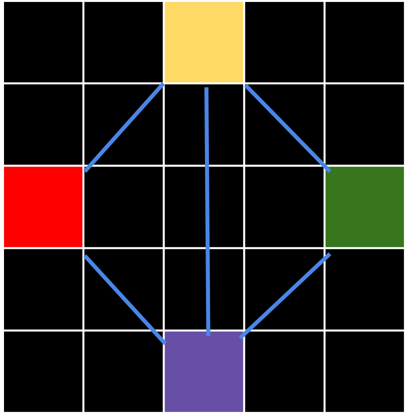
G



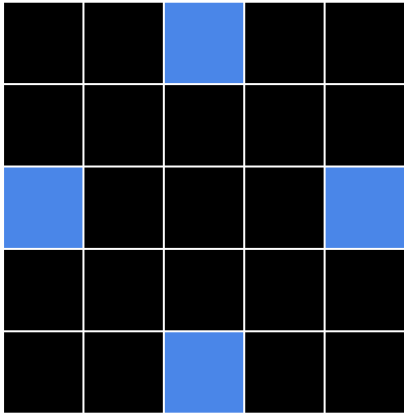
F



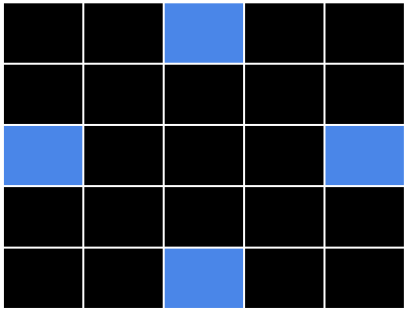
Commitment and example

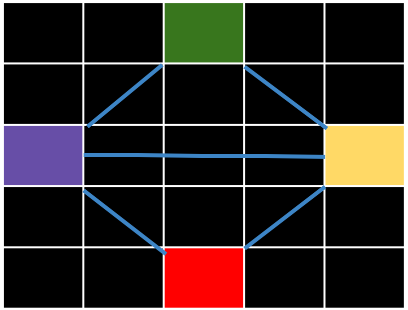


G



F





Isomorphism

Completeness, Soundness

Completeness, Soundness

- Completeness: if P knows a Hamiltonian cycle, V will accept in all cases

Completeness, Soundness

- Completeness: if P knows a Hamiltonian cycle, V will accept in all cases
- Soundness: if P does not know, the best he can do is either create an isomorphic F, or create a Hamiltonian cycle. V will accept 50% of the times -> repeat to pass soundness

Zero Knowledge

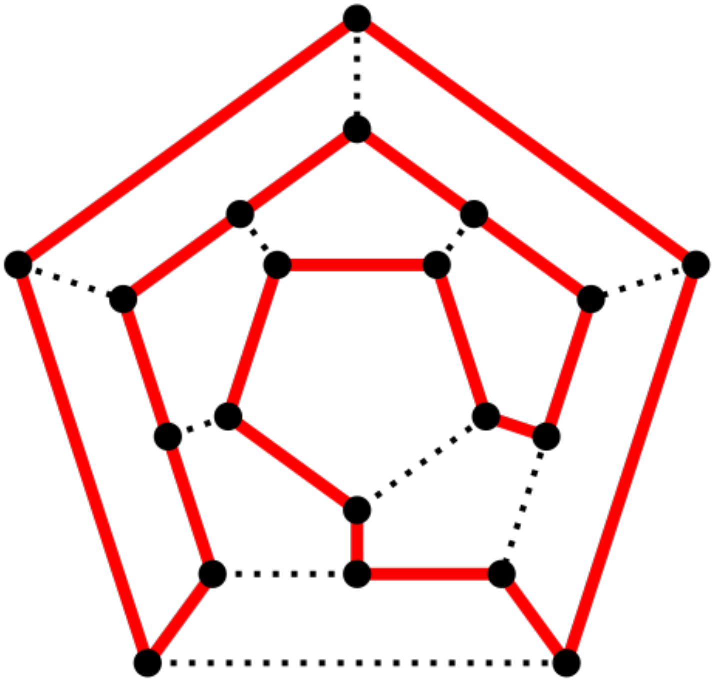
Zero Knowledge

- Suppose V choses ‘isomorphism’. Then all she sees is a ‘scrambled’ version of G. A simulator does not need P to create a random permutation of G

Zero Knowledge

- Suppose V choses ‘isomorphism’. Then all she sees is a ‘scrambled’ version of G. A simulator does not need P to create a random permutation of G
- Suppose V choses ‘cycle’. Then all she sees is a cycle between some n vertices. Since the permutation was random, a simulator that generates random cycles for n vertices would have the same output distribution

Distribution example



Zero Knowledge

- Suppose V choses ‘isomorphism’. Then all she sees is a ‘scrambled’ version of G. A simulator does not need P to create a random permutation of G
- Suppose V choses ‘cycle’. Then all she sees is a cycle between some n vertices. Since the permutation was random, a simulator that generates random cycles for n vertices would have the same output distribution
- V does not learn anything!

Turing Machines (1)

Turing Machines (1)

- Input for both P and V is the graph G (for example represented as a matrix)

Turing Machines (1)

- Input for both P and V is the graph G (for example represented as a matrix)
- P knows a Hamiltonian cycle for G. Uses random tape to create F, isomorphic to G

Turing Machines (1)

- Input for both P and V is the graph G (for example represented as a matrix)
- P knows a Hamiltonian cycle for G. Uses random tape to create F, isomorphic to G
- P commits F using some fancy encryption stuff

Turing Machines (1)

- Input for both P and V is the graph G (for example represented as a matrix)
- P knows a Hamiltonian cycle for G. Uses random tape to create F, isomorphic to G
- P commits F using some fancy encryption stuff
- V randomly selects 1 or 0

Turing Machines (2)

- If 0, P shows the entire committed graph/matrix and how it is isomorphic to G

Turing Machines (2)

- If 0, P shows the entire committed graph/matrix and how it is isomorphic to G
- If 1, P shows the cycle (in the case of a matrix, this means that every row and every column contains two 1s)

Turing Machines (2)

- If 0, P shows the entire committed graph/matrix and how it is isomorphic to G
- If 1, P shows the cycle (in the case of a matrix, this means that every row and every column contains two 1s)
- Verifier checks whether prover is correct

Travelling Salesman

Travelling Salesman

- Famous variant of Hamilton cycle: given a weighted graph (i.e. edges have a certain value), find the *shortest* Hamiltonian cycle

Travelling Salesman

- Famous variant of Hamilton cycle: given a weighted graph (i.e. edges have a certain value), find the *shortest* Hamiltonian cycle
- NP Hard -> Not only check whether the path is a Hamiltonian cycle, but also whether it is the shortest

Holiday

Shortest path through all US towns/cities with more than 500 citizens